The Role of Optional Co-Composition to Solve lexical and Syntactic Ambiguity*

Pablo Gamallo
CITI, FCT, UNL, Portugal
gamallo@fct.unl.pt

Gabriel P. Lopes
CITI, FCT, UNL, Portugal
gpl@di.fct.unl.pt

Alexandre Agustini
PUCRS, Brasil
agustini@inf.pucrs.br

Resumen: Este artículo tiene como objeto estudiar algunas propiedades de lo que llamamos "co-composición opcional". En particular, intenta describir el papel de la co-composición opcional en la desambiguación, tanto del sentido de las palabras como de la estructural. En lo que respecta a esta última, llevaremos a cabo algunos experimentos usando corpus portugués.
Palabras clave: Desambiguación del sentido de las palabras, desambiguación sintáctica, Co-composición, Extracción de información.

Abstract: The aim of this paper is to describe some properties of what we call "optional co-composition". We are interested, more precisely, in the role of optional co-composition in two disambiguation tasks: both word sense and structural disambiguation. Concerning the second task, some experiments were performed on a Portuguese corpus.
Keywords: Word sense disambiguation, Structural disambiguation, Co-composition, Information Extraction

1. *Introduction*

The objective of this paper is to describe the role of binary syntactic dependencies (i.e., "head-dependent" relations) in two specific NLP tasks: both word and structural disambiguation. Our work is mainly based on two assumptions: first, there are two semantic structures underlying a dependency: one where the dependent word is semantically required by the head, and another where the head word is semantically required by the dependent. Second, for some particular tasks such as word disambiguation or syntactic attachment, we also assume that one of both structures can be more discriminant. So, in order to select a word sense or a syntactic attachment, the most discriminant structure will be retained, in particular, the one containing the least ambiguous word. In special cases, for instance when the two related words are highly ambiguous, both structures can be retained.

Consider the two semantic structures underlying the expression "to drive a tunnel". On the one hand, "drive" is a verb that requires, among others, nouns denoting holes or underground passages, and "tunnel" denotes an object satisfying such a requirement. On the other hand, "tunnel" is viewed as a specific predicate requiring verbs denoting *making* events, and "drive" (when it denotes the action of excavating) refers to an event satisfying such a condition. Each predicative structure represents a particular semantic construal of the scene described by the whole expression. This is in accordance with Cognitive Grammar, which claims that an extralinguistic referent can be described by means of various semantic structures, which represent partial viewpoints of that referent (Langacker, 1991). However, we claim, and will show sufficient supporting evidence for this claim, that, in order to analyze and interpret a binary dependency, it is not necessary to use the information associated with the two complementary semantic structures. One of them is enough to obtain an appropriate semantic representation of the dependency. For instance, suppose we are building a computational lexicon by automatically extracting information on selection restrictions from a training corpus. As far as verb "drive" is concerned, corpus co-occurrences may allow the extractor to learn that "drive" requires nouns denoting vehicles: e.g., *cars*, *buses*, *trucks*, etc. However, it may occur that there is insufficient evidence to learn that "drive" requires holes or underground passages like tunnels. This problem arises because "drive" is a very ambiguous word. To

* The work by Pablo Gamallo is supported by the grant of FCT and FSE within the "III Quadro Comunitário de Apoio". The work by Alexandre Agustini is supported by CAPES and PUCRS, Brazil.

overcome this problem, the extractor must be able to learn the complementary structure. That is, it must check if noun "tunnel" requires verbs such as "build", "dig", "excavate", "make", or "drive', which denote *making* events. Satisfying this requirement is enough to posit that "tunnel" is a semantically adequate direct object of "drive". So, we do not need to exhaustively define verb, noun or adjective selection restrictions. Once a unidirectional requirement is learned, it may be used to parse and interpret an expression. We call this phenomenon *optional co-composition*.

The main contribution of this paper is to define some properties of optional co-composition as well as how it is involved in two disambiguation tasks: word sense disambiguation and attachment resolution (syntactic structure disambiguation).

This paper is organized as follows. First, section 2 introduces the most common ideas concerning the semantic structure of binary dependencies. Then, in section 3.2 we will make new assumptions for what we consider to be the restrictive structure of a dependency. This structure will be defined on the basis of the notion of optional co-composition. According to this notion, a word, despite of its syntactic role in a dependency, may semantically restrict (select) the words with which it can combine. We will analyse also co-compositionality within the Generative Lexicon Theory. In these two sections, the description will be focused on a particular task: word sense disambiguation. Finally, in section 4, we will focus on a different task: the acquisition of selection restrictions from corpora, and the use of selection restrictions to solve structural ambiguity. Some empirical results will be given.

2. *Disambiguation Using a Standard Predicative Structure*

In linguistic theories based on Dependency Grammar (Hudson, 2003), predicate-argument structures are seen as semantic representations of head-dependent syntactic dependencies. The semantic structure of a dependency is constituted by the word that imposes linguistic constraints (the predicate) and the word that must fill such constraints (its argument). Each word is supposed to play a fixed role. The argument is perceived as the word satisfying the constraints imposed by the predicate, while the predicate is viewed as the active selector imposing constraints on the former. Typically, the role of a word in a predicate-argument structure is fixed by its morphosyntactic category: for instance, verbs and (some) adjectives are taken as predicates while nouns are seen as their arguments. Most model-theoretical approaches propose semantic representations of word denotations that are in accordance with these ideas. Verbs and adjectives denote functional objects that take as arguments denotations of nominals.

However, some shortcomings can arise from a fixed predicate-argument structure. Its rigidity does not allow the argument to impose requirements on the predicate and, then, it makes the disambiguation process less efficient. Take the expression:

$$\text{long dinner} \tag{1}$$

According to the definitions provided by most Machine Readable Dictionaries (MRD) and lexical resources such as WordNet 2.0, both words are polysemous. Word "dinner" is, in general, defined as having two senses. It denotes a temporal event ("the dinner will be at 8"), and a mass object without dimensions, i.e., the food itself ("light dinner"). On the other hand, adjective "long" seems to be more polysemous. In WordNet 2.0, it has 10 senses: e.g., a primarily temporal dimension, a primarily spatial dimension, a phonetic property, and so on. Considering the fixed predicate-argument structure associated with expression (1), the adjective must be taken as the functional predicate and the noun as the argument. So, selection restrictions are only imposed by "long" on the noun.

Let us propose a disambiguation strategy based on a fixed predicative structure. This strategy would consist of the following steps:

Enumerating the senses of the predicate:
The 10 senses of the adjective "long", which are represented as 10 different lexical predicates, are put in a list. In Figure 1, this list is the column of predicates associated to "long". They are enumerated from 1 to 10. Attributes 'space', 'time', etc. represent the selection restrictions of each predicate.
Checking restrictions:
Each predicate is applied to the senses of "dinner", which are also represented by

attributes: 'mass' and 'time'. Each functional application checks whether there is one sense of "dinner" satisfying the requirements of the predicate. This process is repeated until there are no more predicates to check.

Selecting senses:

Each time that the requirements are satisfied by a sense of the noun, a predicate is selected. In this example, only one functional application is allowed by the semantic requirements: the one selecting the *time* sense (application 2 in Figure 1) of "long" and "dinner". The final result is an interpretable formula:

$$Long(dinner : time) \quad (2)$$

This strategy makes the process of word disambiguation very rigid. Only the semantic requirements of the predicate allow to select for the appropriate sense of the argument. Yet, the argument is not associated with requirements likely to be used to select for the appropriate predicate. The selection of the correct predicate is not driven by the argument restrictions; it is simply done by enumerating one by one the different senses of the word taken as predicate. Predicative enumeration turns out to be highly inefficient if the predicate is associated to the most polysemous word within a dependency. We argue that enumerating all possible senses of the most ambiguous word is not the more suitable way to simulate the understanding process. According to psychological evidences (Justeson y Katz, 1995), it seems to be more effective to allow the least ambiguous word to be the active disambiguator (i.e., the sense discriminant).[1]

In expression (3), there is a more radical difference concerning the degree of polysemy conveyed by two words:

$$\text{to drive the tunnel} \quad (3)$$

In WordNet, "drive" has 21 senses; one of them represents the event of making a passage by excavating. By contrast, "tunnel" merely has 2 much related senses. In order to interpret expression (3), we argue that

[1]Note that this approach to the disambiguation process is not dependent on a particular notion on word sense. The same problems arise even if more radical notions on word sense are considered: e.g., a sense taken as a set of words with similar context distribution (Schütze, 1998).

1. $[\lambda x\, Long(x : space)] \quad (dinner : \begin{bmatrix} mass \\ time \end{bmatrix})$
2. $[\lambda x\, Long(x : \mathbf{time})] \quad (dinner : \begin{bmatrix} mass \\ \mathbf{time} \end{bmatrix})$

   $\vdots$

10. $[\lambda x\, Long(x : vowel)] \quad (dinner : \begin{bmatrix} mass \\ time \end{bmatrix})$

Figura 1: Application of predicate $\lambda x Long(x)$ to argument *dinner*.

the hearer/reader uses the least ambiguous nominal argument as disambiguator. It is the noun that selects for a specific verb sense: the *making* sense. This disambiguating process is psychologically and computationally supported. On the other hand, checking one by one each sense of the verb requires access to multiple entries of a specific lexicon, assumes that all senses and all selection restrictions are represented, and worst than that, does not allow a more efficient disambiguation procedure, where the argument might play the role of an active disambiguator.

3. Disambiguation Using Co-composition

We assume that the two words related by a syntactic dependency impose semantic restrictions on each other. Not only verbs or adjectives must be taken as predicates selecting different senses of noun denotations, but also nouns should be considered as active functors requiring different senses of verb and adjective denotations.

3.1. Co-Composition in Generative Lexicon

Generative Lexicon (GL) proposes that nominal complements carry lexical information used to shift the sense of the governing verb (Pustejovsky, 1995). It means that GL allows some nominal arguments to constrain the meaning of their verbal predicates. However, the functional application used by GL relies on the classical approach: relational words (verbs and adjectives) are taken as functions, while nominals are taken as being their arguments. Co-composition is viewed here as a unification operation that restricts function application. This operation is triggered off only if both the verb and the noun

contain vey specific lexical information. The scope of this particular operation is then very narrow. We claim, however, that co-composition is a general semantic property underlying every syntactic dependency between two words. In the next subsection, we will generalise the notion of co-composition so as to deal with all cases of word dependencies. To do it, functional application will not be driven by relational words, but by dependencies.

3.2. Optional Co-composition

We consider dependencies as active objects that control and regulate the selection requirements imposed by the two related words. So, they are not taken here as merely passive syntactic cues related in a particular way (linking rules, syntactic-semantic mappings, syntactic assignments, etc.) to thematic roles or lexical entailments of verbs (Dowty, 1989). They are conceived of as the main functional operations taking part in the process of sense interpretation.

On this basis, we associate functional application, not to relational expressions (verbs, adjectives, ...), but to dependencies. In functional terms, a dependency can be defined as a binary λ-expression:

$$\lambda x \lambda y \, dep(x, y) \tag{4}$$

where x and y are variables for word meanings. The meaning of the head word, x, will be in the first position, while the meaning of the dependent, y, will be in the second one. The different types of dependency we consider are the following: nominal verb complement situated to the left of the verb ($lobj$), or to the right of the verb ($robj$), prepositional complement of the verb ($iobj\_prep - name$), prepositional complement of the noun ($prep-name$), and attributive function of the adjective ($attr$).

The objective of this subsection is to show how dependencies can be used to disambiguate words in a co-compositional way. To do it, let's take again expression (1) (i.e., "long dinner"). The word disambiguation strategy we propose here consists of the following steps:

Identifying a dependency function:
From the adjective-noun expression, the $attr$ binary function is proposed:

$$\lambda x \lambda y \, attr(x, y) \tag{5}$$

Choice of a word disambiguator:
The dependency function is applied first to the word considered to be the best discriminator. By default, it will be the word with the least number of senses, that is, the least polysemous word. As has been said before, the chosen word must be "dinner". As a result, this word is assigned to the head position of dependency $attr$:

$$\begin{aligned} &[\lambda x \lambda y \, attr(x, y)] \; (dinner) \\ &\lambda y \, attr(dinner, y) \end{aligned} \tag{6}$$

This is still a predicative function likely to be applied to the word in the dependent position. Consequently, word "dinner", in the head position, is taken here as the active predicate. Basically, the remainder steps are the same as in the previous strategy.
Enumerating the senses of the predicate:
The 2 senses of noun "dinner" are put in a list of predicates (see Figure 2). The predicates are enumerated from 1 to 2, with their respective selection restrictions: $time$ and $mass$.
Checking restrictions:
Each predicate is applied to the meaning of "long". Here, only two checking operations are activated.
Selecting senses:
The requirements imposed by the temporal predicate allow to select the temporal sense of the adjective. Functional application gives rise to an interpretable formula:

$$attr(dinner : time, long : time) \tag{7}$$

where the two related words denote compatible senses. Such a procedure is independent of the way we represent (as features, word clusters, probabilities, etc.) word senses and selection restrictions.

This strategy is more efficient than that defined in the previous section, since here the disambiguation process is controlled by the word that is considered to be the most appropriate to discriminate the sense of the other one. Moreover, optional co-composition makes functional application more flexible, since it allows to choose as predicative function whatever word within a dependency, or even, if necessary, both words. Any word of a binary dependency may become the lexical function and, then, be used to disambiguate the meaning of the other word.

1. $[\lambda y\ \text{attr(dinner, y:mass)}]$ (long: $\begin{bmatrix} \text{space} \\ \text{time} \\ \vdots \\ \text{vowel} \end{bmatrix}$)

2. $[\lambda y\ \text{attr(dinner, y:}\mathbf{\text{time}}\text{)}]$ (long: $\begin{bmatrix} \text{space} \\ \mathbf{\text{time}} \\ \vdots \\ \text{vowel} \end{bmatrix}$)

Figura 2: Application of predicate $\lambda y attr(dinner, y)$ to argument *long*

Nevertheless, word disambiguation should not be restricted to a single binary dependency. The target word is actually disambiguated by all words to which it is syntactically related. So, the disambiguating context of a word is not only a single dependency, but also the set of dependencies it participates in. This remains beyond the scope of the paper.

We have described in this section the internal structure of syntactic dependencies and how they can be used to disambiguate words in a flexible way. In the following section, we will see the benefits of optional co-composition in a different task: syntactic disambiguation.

4. *Using Co-composition to Solve Syntactic Ambiguity*

This section describes a method to solve syntactic attachment. First, we acquire selection restrictions from corpora, then the acquired information is used to build a subcategorization lexicon. Finally, a specific heuristic is used to propose correct syntactic attachments. The main characteristic of the method is the use of the assumption on optional co-composition introduced in the previous section. Some results are evaluated at the end of the section. This method has been accurately described in (Gamallo, Agustini, y Lopes, 2003).

4.1. Selection Restrictions Acquisition

An experiment to automatically acquire selection restrictions was performed ona Portuguese corpus[2]. We used an unsupervised and knowledge-poor method. It is unsupervised because no training corpora semantically labeled and corrected by hand is needed. It is knowledge-poor since no handcrafted thesaurus such as WordNet nor no MRD is required (Grefenstette, 1994).

The method consists of the following steps. First, raw Portuguese text is automatically tagged and then analyzed in binary syntactic dependencies using a simple heuristic based on Right Association. For instance, the expression "the salary of the secretary"gives rise to the relation:

$$of(salary, secretary) \tag{8}$$

Then, following the assumption on co-composition, we extract two different functional predicates from every binary dependency. From (8), we extract:

$$\lambda y\ of(salary, y) \tag{9}$$
$$\lambda x\ of(x, secretary) \tag{10}$$

Finally, we generate clusters of predicates by computing their word distribution. We assume, in particular, that different predicates are considered to impose the same selection restrictions if they have similar word distribution. Similarity is calculated by using a particular version of the Lin coefficient (Lin, 1998). As a result, a predicate like (9) may be aggregated into the following cluster:

$$\begin{aligned} &\lambda y\ of(salary, y) \\ &\lambda y\ of(post, y) \\ &\lambda y\ lobj(resign, y) \\ &\lambda x\ attr(x, competent) \end{aligned} \tag{11}$$

which is associated to those words co-occurring at least once with each predicate of the cluster, e.g.:

```
secretary, president,
minister, manager, worker,
journalist
```

We use these words to extensionally define the selection restrictions imposed by the similar predicates of cluster (11). In fact, the set of words required by similar predicates represents the extensional description of their semantic preferences.

[2] 3 million words belonging to the P.G.R. (*Portuguese General Attorney Opinions*) corpus, which is constituted by case-law documents.

4.2. Building a Subcategorization Lexicon

The acquired clusters of predicates and their associated words are used to build a lexicon with syntactic and semantic subcategorization information. Table 1 shows an excerpt of the information learned concerning the entry `secretário` (*secretary*). This entry defines six different predicative structures. Notice that it is the notion of co-composition that allows us to define a great number of predicates that are not usual in the standard approaches to subcategorization. Five of the six predicates with *secretary* do not subcategorize standard dependent complements, but different types of heads. This is a significant novelty of our approach.

4.3. Attachment Heuristic

Optional co-composition is also at the center of syntactic disambiguation. It underlies the heuristic we use to check if two phrases are dependent or not. This heuristic states that two phrases are syntactically attached only if one of these two conditions is verified: either the *dependent* is semantically required by the *head*, or the *head* is semantically required by the *dependent*. Take the expression:

`compete ao secretário do ministro`
(*is incumbent on the secretary of the minister*) (12)

There exist at least three possible attachments: 1) `competir` (*be incumbent*) is attached to `secretário` by means of preposition `a`; 2) `competir` is attached to `ministro` by means of preposition `de`; 3) `secretário` is attached to `ministro` by means of preposition `de`. Each attachment is verified using the co-compositional information stored in the lexicon. For instance, the first attachment is verified if only if, at least, one of the two following conditions is satisfied:

Dependent Condition: $\lambda y\ iobj\_a(competir, y)$ subcategorizes a class of nouns to which `secretário` belongs;

Head Condition: $\lambda x\ iobj\_a(x, secretário)$ subcategorizes a class of verbs to which `competir` belongs.

According to the lexical information illustrated in Table 1, the attachment is allowed because the Head Condition is satisfied by the verb. Note that, even if we had not learned information on the verb restrictions, the attachment would be allowed since the restrictions imposed by one of the two possible predicative structures (the nom-

secretário (*secretary*)

· $\lambda x\ of(x, secretário)$ =
`cargo, carreira, categoria, competência, escalão, estatuto, funçao, remuneracão, trabalho, vencimento`
(post, career, category, qualification, rank, status, function, remuneration, job, salary)

· $\lambda y\ of(secretário, y)$ =
`administração, assembleia, autoridade, conselho, direcção, empresa, entidade, estado, governo, instituto, juiz, ministro, ministério, presidente, serviço, tribunal órgão`
(administration, assembly, authority, council direction, company, entity, state, government, institute, judge, minister, ministery, president, service, tribunal organ)

· $\lambda x\ iobj\_a(x, secretário)$ =
`aludir, aplicar:refl, atender, atribuir, concernir, corresponder, determinar, presidir, recorrer, referir:refl, respeitar`
(allude, apply, attend, assign, concern, correspond, determine, resort, refer, relate)

· $\lambda x\ iobj\_a(x, secretário)$ =
`caber, competir, conceder:vpp, conferir, confiar:vpp, dirigir, incumbir, pertencer`
(concern, be-incombent, concede, confer, trust, send, be-incombent, belong)

· $\lambda x\ iobj\_por(x, secretário)$ =
`assinar, conceder, conferir, homologar, louvar, subscrever`
(sign, concede, confer, homologate, compliment, subscribe)

· $\lambda x\ lobj(x, secretário)$ =
`definir, estabelecer, fazer, fixar, indicar, prever, referir`
define, establish, make, fix, indicate, foresee, refer

Cuadro 1: Excerpt of lexicon entry `secretário`

inal one) are satisfied. Following this attachment procedure, we are able to decide that `secretário` and `ministro` are dependent, but not `competir` and `ministro`.

4.4. Evaluation

Table 2 reports the test scores concerning the precision and recall of the experiments performed on 1266 test expressions, which have been selected randomly from a test corpus. As (12), each test expression contains three phrases and, then, three possible attachments. We made a comparison between our method and a baseline strategy. As a baseline, we used the attachments proposed by Right Association. That is, for each expression of the test data, this strategy always proposes the attachment by proximity, that is: *phrase1* is attached to *phrase2*, *phrase2* is attached to *phrase3*, and *phrase1* is not attached to *phrase3*.

Each expression selected from the test corpus contains three phrases and three candidate attachments. So, given a test expression, three different attachment decisions will be evaluated. The evaluation of each attachment decision taken by the system can be:

- true positive (*tp*): the system proposes a correct attachment
- true negative (*tn*): the system proposes correctly that there is no attachment.
- false positive (*fp*): the system proposes an incorrect attachment
- false negative (*fn*): the system proposes incorrectly that there is no attachment.

The evaluation test measures the ability of the system to make true decisions. We call both *tp* and *tn* "true decisions" (*td*). As far our strategy is concerned, a false negative (*fn*) is interpreted as the situation in which the system has not enough subcategorization information to make a decision. Concerning the baseline, the *fn* decisions correspond to those situations where there is a true long distance attachment: *phrase1* is attached to *phrase3*.

Taking into account these variables, *precision* is defined as the number of true decisions suggested by the system divided by the number of total suggestions. That is:

$$precision = \frac{td}{td + fp} \tag{13}$$

Recall is computed as the number of true decisions suggested by the system divided by the decisions that are actually correct:

$$recall = \frac{td}{td + fn} \tag{14}$$

Note that, in our evaluation protocol, the test expressions are not only (vp-np-pp) sequences of phrases, as in most work on attachment disambiguation. Test data was separated in three groups of expressions according to three different syntactic sequences: (vp-np-pp), (vp-pp-pp), and (np-pp-pp).

The values plotted in Table 2 shows that the results performed using optional co-compositional information are significatively better concerning precision. As far as recall is concerned, our approach remains lower than the baseline since the subcategorization lexicon is still incomplete. In order to improve recall, we need to allow the clustering strategy to generate more general classes. This will provide the lexicon with more items of subcategorization information.

These results can be hardly compared to related approaches given that there is no related work on Portuguese. Moreover, we use three types of phrase sequences, and not only the (vp-np-pp) sequence used by most related work.

BASELINE			
sequences	**Prec.** (%)	**Rec.** (%)	**F-S** (%)
np,pp,pp	71	71	71
vp,pp,pp	83	83	83
vp,np,pp	75	75	75
Total	76	76	76
OPTIONAL CO-COMPOSITION			
sequences	**Prec.** (%)	**Rec.** (%)	**F-S** (%)
np,pp,pp	85	73	79
vp,np,pp	92	75	83
vp,pp,pp	86	70	77
Total	88	73	80

Cuadro 2: Evaluation of Attachment Resolution

5. Conclusion

This paper has introduced a particular property of syntactic dependencies, namely optional co-composition, and its role in the process of disambiguation. This property allows learning two complementary semantic

structures of a dependency, even if only one of them contains enough information to select a word sense or a specific syntactic attachment.

The theoretical background underlying many works on NLP is often far from most recent and innovative approaches to lexical semantics, cognitive linguistics, or other linguistic areas. The main contribution of the paper is to merge different theoretical approaches (generative lexicon and cognitive grammar) in order to define a sound notion, optional co-compositionality, and describe how it can be used in different NLP applications. In sum, our aim is to use some ideas taken from current linguistic approaches to improve NLP applications.

Bibliografía

Dowty, D.R. 1989. On the semantic content of the notion of thematic role. En *Properties, Types, and Meaning, vol. 2*. Kluwer Academic Publisher, páginas 69–130.

Gamallo, Pablo, A. Agustini, y Gabriel P. Lopes. 2003. Learning subcategorisation information to model a grammar with co-restrictions. *Traitement Automatic de la Langue*, 44(1):93–117.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, USA.

Hudson, Richard. 2003. The psychological reality of syntactic dependency relations. En *MTT2003*, Paris.

Justeson, John y Slava Katz. 1995. Principled disambiguation: Discriminating adjective senses with modified nouns. *Computational Linguistics*, 21(1):1–27.

Langacker, Ronald W. 1991. *Foundations of Cognitive Grammar: Descriptive Applications*, volumen 2. Stanford University Press, Stanford.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. En *COLING-ACL'98*, Montreal.

Pustejovsky, James. 1995. *The Generative Lexicon*. MIT Press, Cambridge.

Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.